Ames Research Center

Integration of On-board EOS Schedule Revision with Space Communications Emulation System

Lina Khatib and Robert Morris (NASA/ARC)
Rich Slywczak and Thong Luu (NASA/GRC)

ESTC: June 23, 2004

A Follow Up work to ESTO/AIST funded project completed last year!

Current funding: CICT/IS Program.

Presentation Plan

- EOS scheduling problem
- On-board schedule revision (needs and approach)
- Space communications (needs and emulation)
- Integration of On-board schedule revision and Space Communications using SCEF
- Conclusions

Observation Scheduling for Earth Orbiting Satellites

Ames Research Center

- Given a set of requests for images of the Earth, a set of sensing instruments, and a set of constraints, produce a set of assignments of instruments and viewing times to those requests that satisfy the constraints.
- Constraints associated with EOS Scheduling
  - On-board storage (Solid State Recorder) capacity
  - Instrument duty cycle
  - Slewing (for agile instruments)
- Requests associated with scientific utility
  - Importance in meeting science goals
  - Expected utility given viewing conditions (cloud cover)
- Instruments are oversubscribed; more requests than can be serviced.
- Objective: maximize the sum of the utility of requests put on schedule

Observation Scheduling: Current Practice

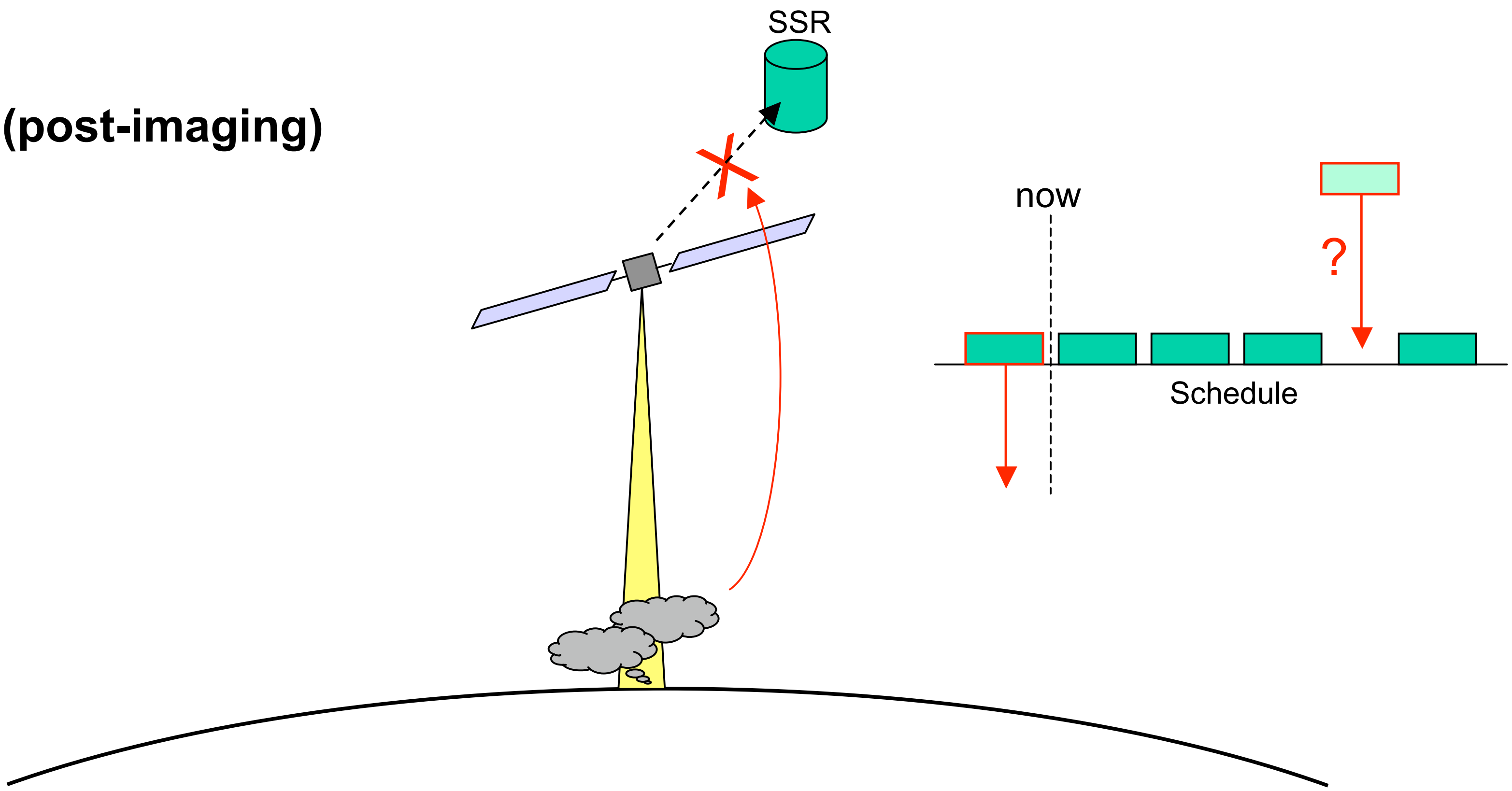
- Performed on the ground for periods covering a day or more
- Command sequences uplinked and executed rigorously
- Utility calculations integrated into scheduling process
- Example: Landsat 7 scheduler [Potter & Gasch, 1998]

Argument for On-board Decision-making

- Relative utility of observation can change dynamically
  - Unexpected cloud cover
  - Serendipitous events
  - Changes in resource capabilities
    - Loss of ground station
    - On-board storage
- Satellites can only communicate with ground occasionally
  - Thus, it may be infeasible to generate desired schedule changes on the ground and uplink them.
- Thus, to maximize utility of acquired images, do some of the decision-making on-board.

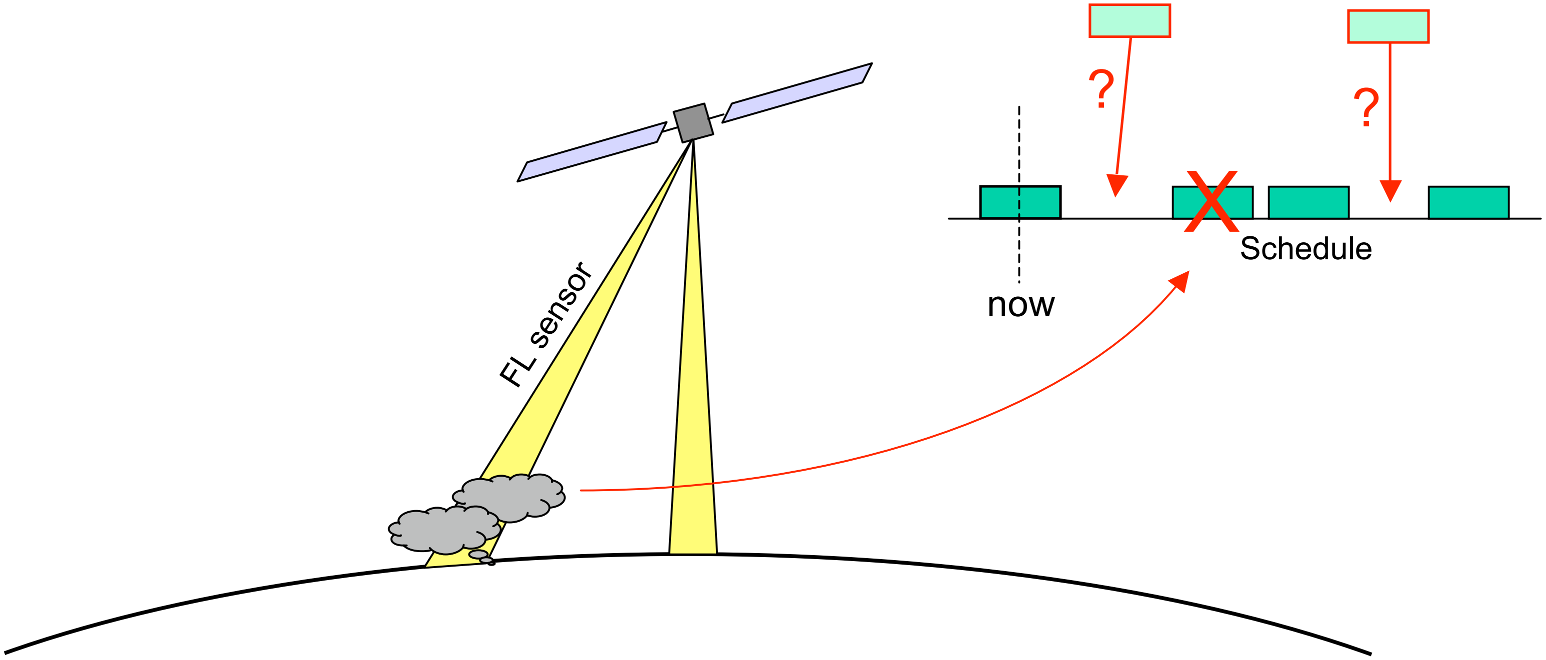
Example: On-board analysis of acquired images may lead to freeing SSR space

(post-imaging)



Example: Forward looking sensors may lead to revising future observations

(Pre-imaging)



Ames Research Center

On-board Schedule Revision

- Approach
  - Complete schedule is initially constructed on the ground
  - A greedy algorithm for on-board schedule revision
  - Ground schedule bias is applied
  - Lookahead strategies applied at each decision step
- Goal: study expected gain in overall science utility as a result of performing on-board schedule revision over rigorous execution of schedule produced on the ground.

On-board Schedule Revision (Experiments)

- Problem instances of 9 hour scheduling horizons and up to 400 requests
- Various frequency and types of dynamic utility revisions
- Various SSR capacity
- Various number of alternative observations considered
- Various biases tactics and lookahead strategies.
- Single/multiple instruments

On-board Schedule Revision (Experimental Results)

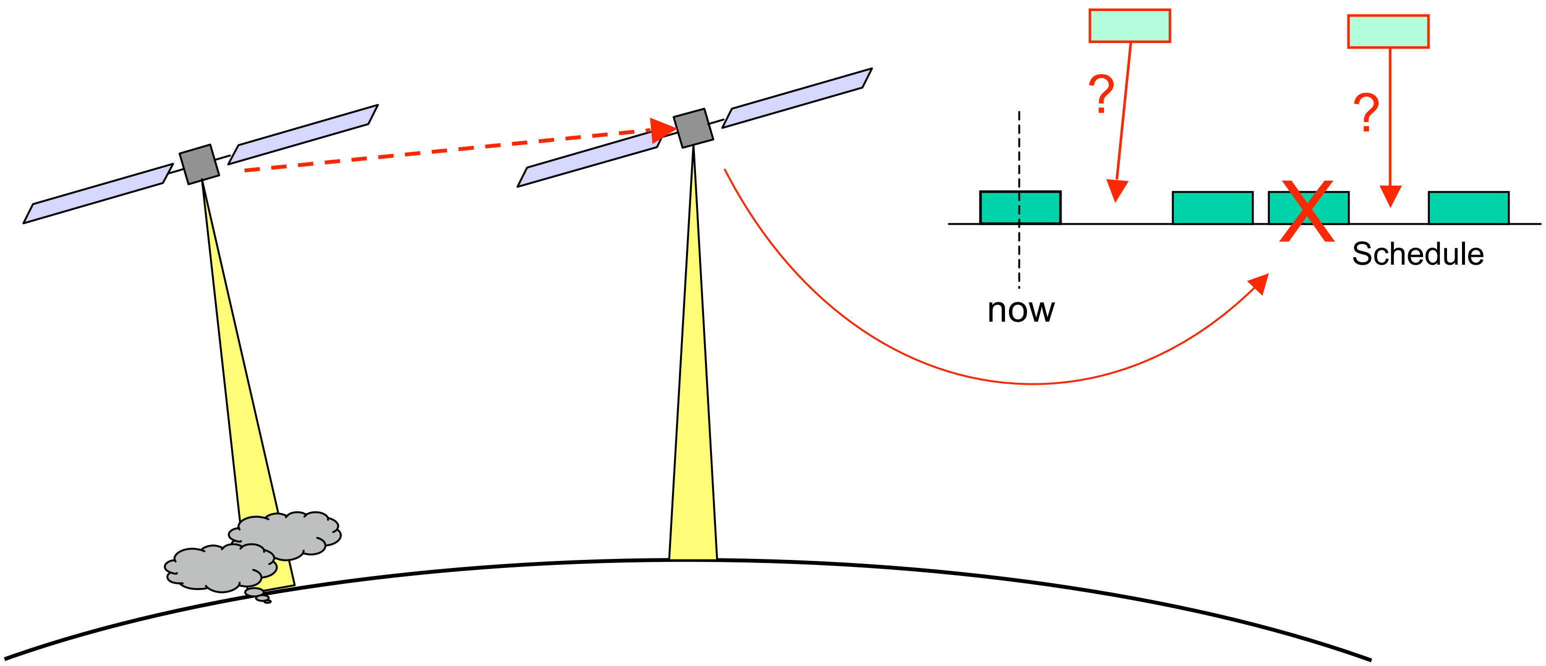
- Revision works better than no revision in terms of overall utility of downloaded images (up to 14 % improvement).
- With refinements to lookahead strategies, pruning techniques cut down on the size of search and, as a result, solutions were found at a reasonable time.
- The interesting phenomena of “horizon effect” was observed with fixed lookahead strategy.
- Variable lookahead works better than fixed lookahead strategy.

Future Space Communications

- NASA is designing more complex missions with stringent communications and coordination requirements.
- Trend is to move from single satellite missions toward multiple satellite missions.
- Example Future Missions
  - Loosely coupled constellations
    - Little communication between each of the satellites.
    - Global Precipitation Mission (GPM)
  - Tightly coupled constellations
    - Communications and coordination is essential among satellites.
    - Micro-Arcsecond X-Ray Imaging (MAXIM)
  - Sensor Web

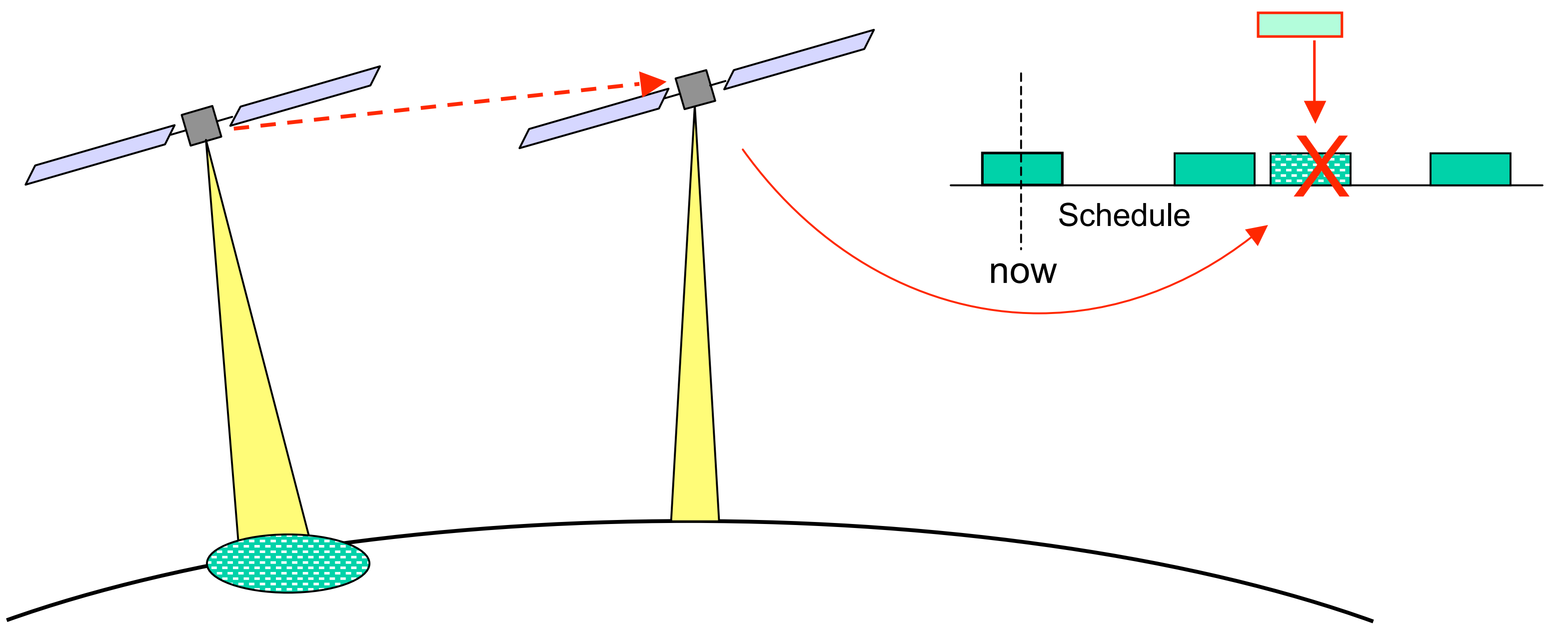
Integrating Space Communications and On-board Schedule Revision

Example: Leading Satellite Detection of Unexpected Cloud (or Target of Opportunity)



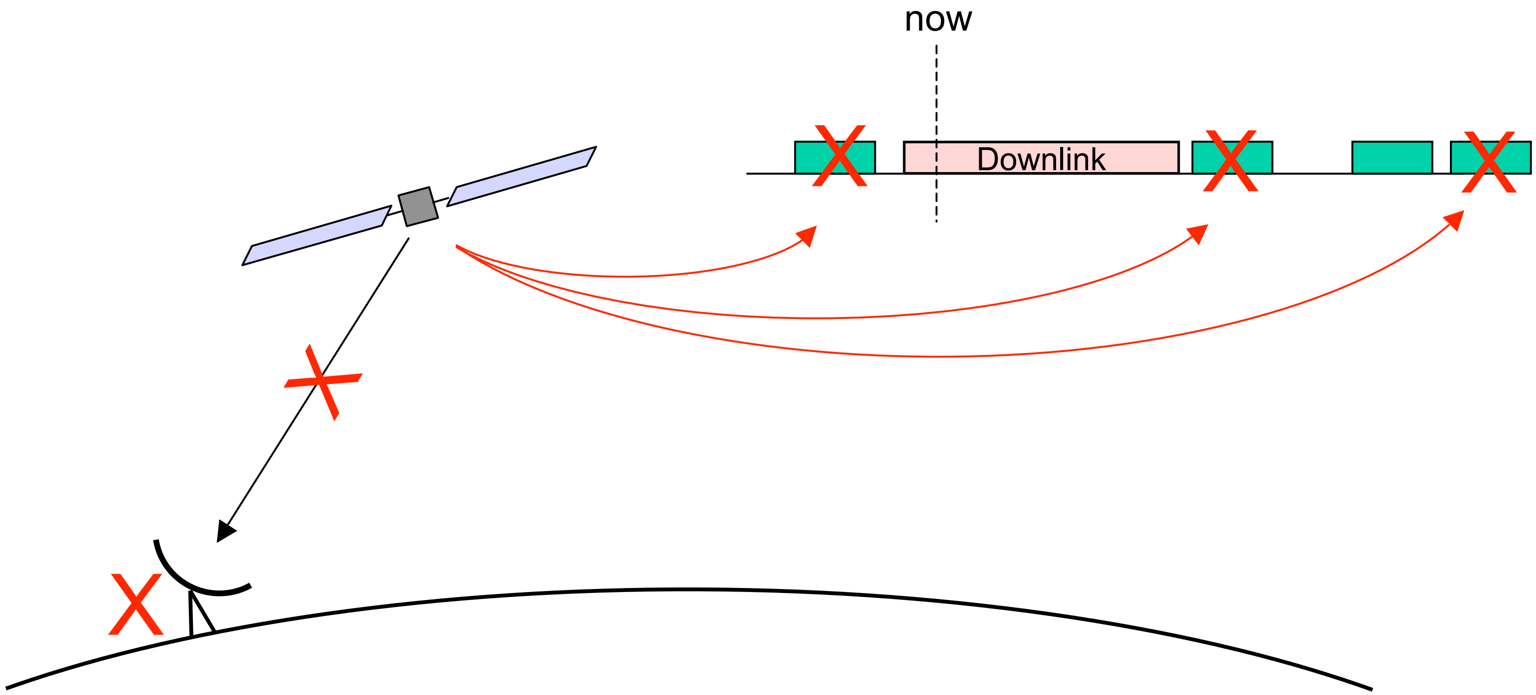
Example: Shared Observations

Replace future observation
Captured by a leading satellite.



Example: Ground (or Relay) Station Loss

Discard data
& future observations



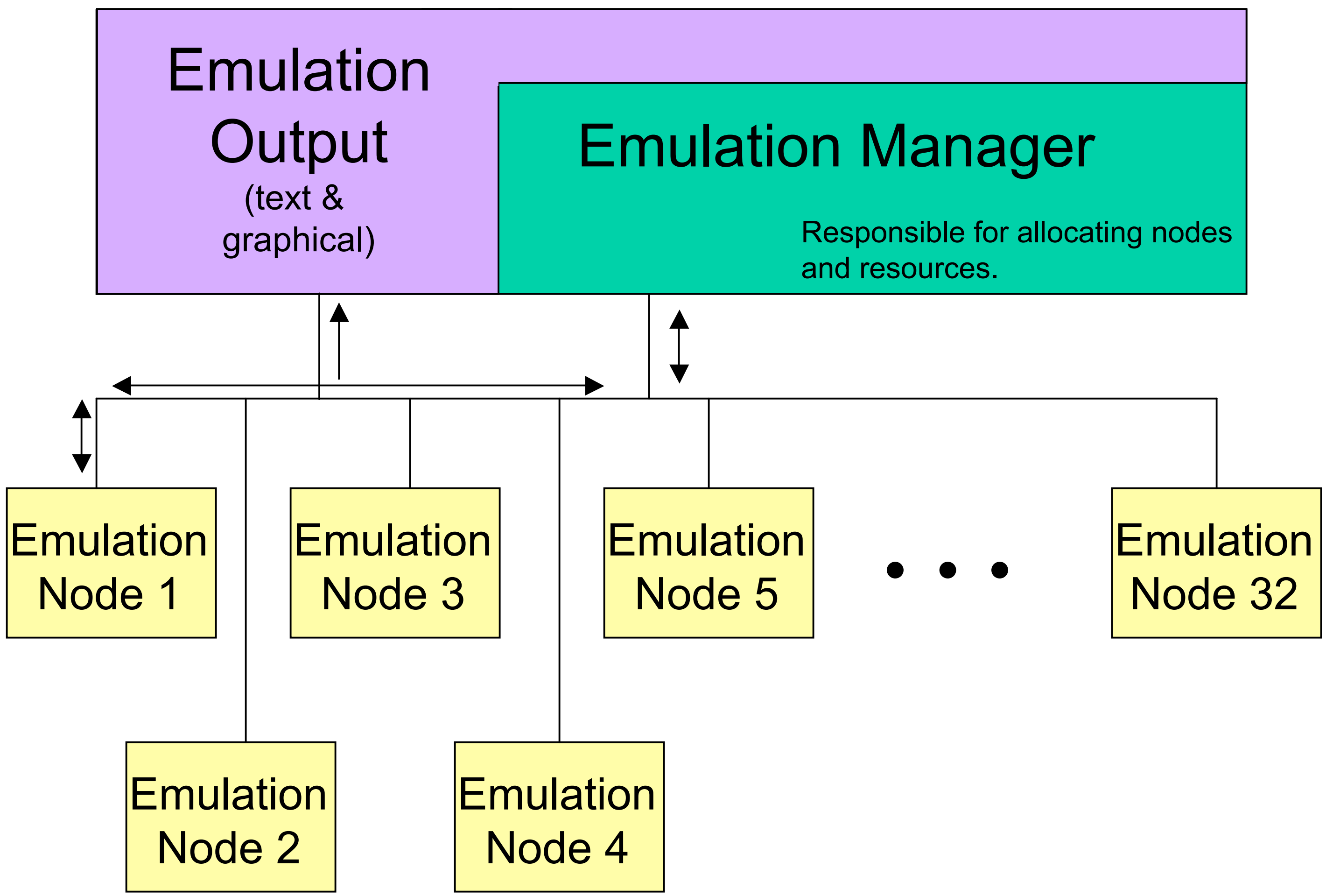
ESTC: June 23, 2004

Space Communications Emulation Facility (SCEF)

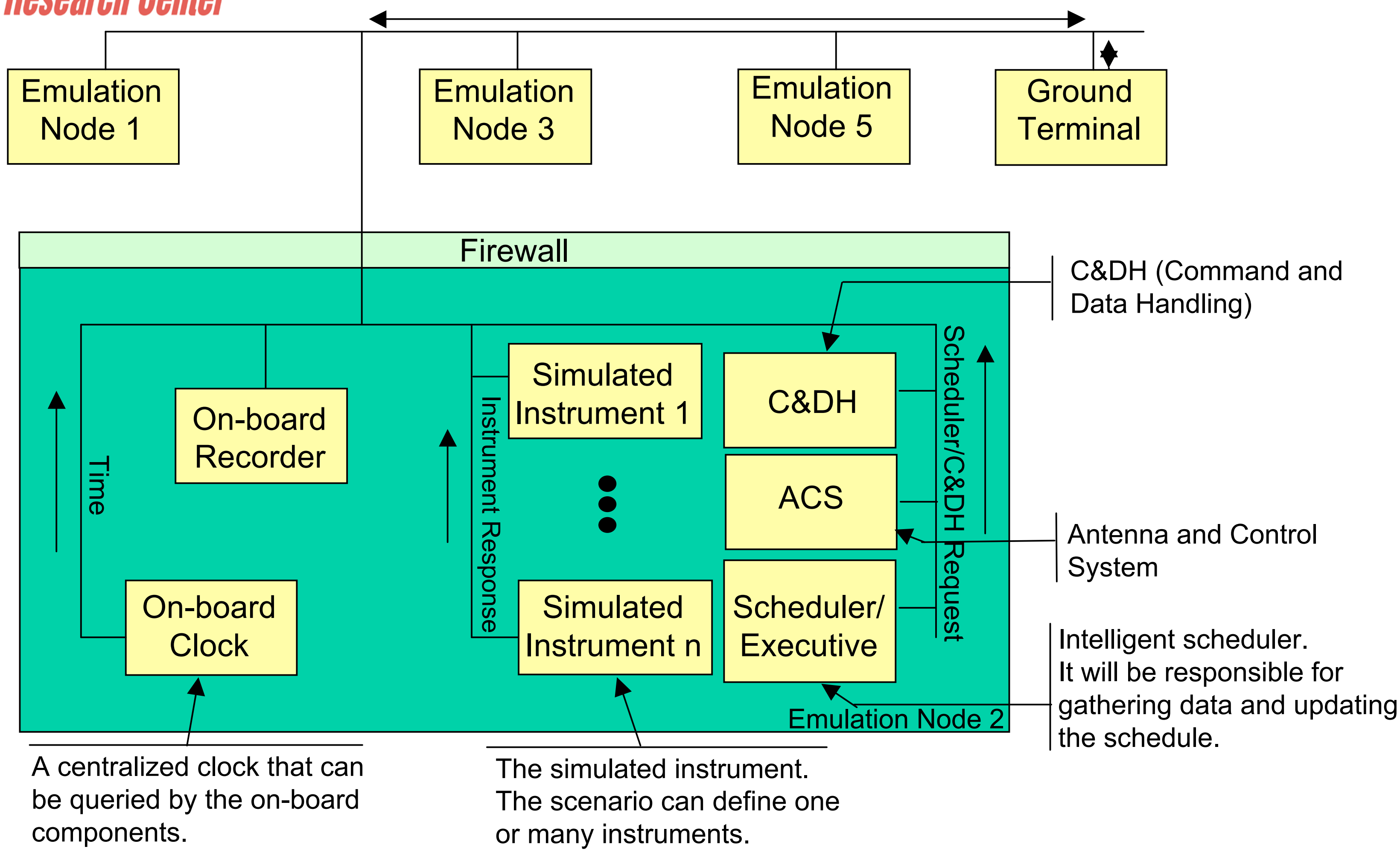
Ames Research Center

- Provide an environment that allows researchers to emulate space missions and/or custom on-board components.
- Visualize the orbits and communications links.
- Provide the ability to add custom codes to the emulation system.
- End users can customize various parameters:
  - Number of satellites
  - Number of instruments on the satellite
  - Orbital Parameters
  - Space environment characteristics (e.g., latency, BERs).
- Originally based on University of Kansas' Space Based Internet (SBI) software.
- Testbed resides at the Glenn Research Center (GRC)
  - Eventually to become a distributed testbed.

SCEF Architecture



SCEF Node Architecture



Theory Meets Reality!

- SCEF is used to emulate the On-board schedule revision interleaved with execution system.
- Real-time computational aspects are being considered:
  - Decision-making overhead time
  - Communication (update) delays
  - Possible execution failures
- Other technical issues:
  - Commitment Window
  - Freeze Time
  - Failure Recovery

Conclusions

- New approach to managing EOS science scheduling based on combined scheduling + limited on-board autonomy.
- Motivation is increasing demand for high quality science data.
- Results suggest improvement in science gain with limited computing resources.
- Joined effort between ARC and GRC uses SCEF as a cost-effective, yet robust, experimental platform.
  - Integration effort is currently at an early stage of development.
  - Real-time execution issues are being tackled.
  - Well-defined test scenarios are on the agenda.

The End!

Introduction

Goal: Provide an environment that allows researchers to emulate space missions and/or custom on-board components.

End users can customize various parameters:

- Number of satellites
- Number of instruments on the satellite
- Orbital Parameters
- Space environment characteristics (e.g., latency, BERs).

Customize the components on-board the satellite

- Provide the ability to add custom codes to the emulation system.

Output can be shown textually and graphically

- Visualize the orbits and communications links
- Text output will show the throughput for the links

Originally based on University of Kansas' Space Based Internet (SBI) software.

Testbed resides at the Glenn Research Center (GRC)

- Eventually to become a distributed testbed.

Motivation

NASA is designing more complex missions with stringent communications and coordination requirements.

Trend is to move from single satellite missions toward multiple satellite missions.

Example Future Missions

- Loosely coupled constellations
  - Little communication between each of the nodes.
  - Global Precipitation Mission (GPM)
- Tightly coupled constellations
  - Communications and coordination is essential among nodes.
  - Micro-Arcsecond X-Ray Imaging (MAXIM)
- Sensor Web

Current testbed focus is Low Earth Orbit (LEO) missions

- Future development will include Lunar and Deep Space Missions.

Advantages of SCEF

Common Infrastructure

- Projects can share results and data from the emulation.
- Promotes more interaction between projects during design.

Space Characteristics

- Implements latency, Bit Error Rates (BERs), QoS, etc.

Satellite Components

- Provides default algorithms for C&DH, ACS, Instruments, on-board clock, etc.

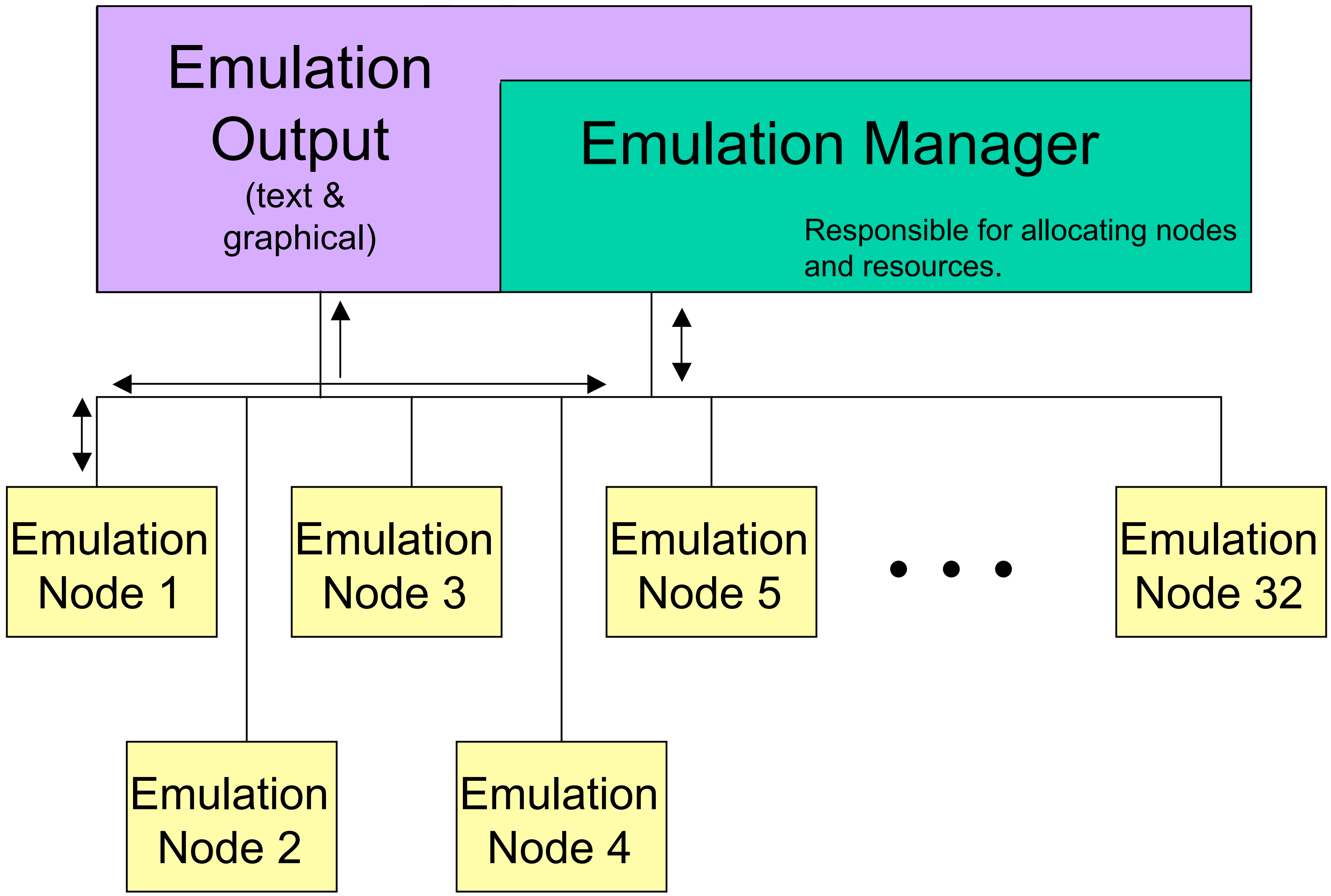
Cost Reduction

- Multiple use of common software.
- Evaluate missions and concepts during design.

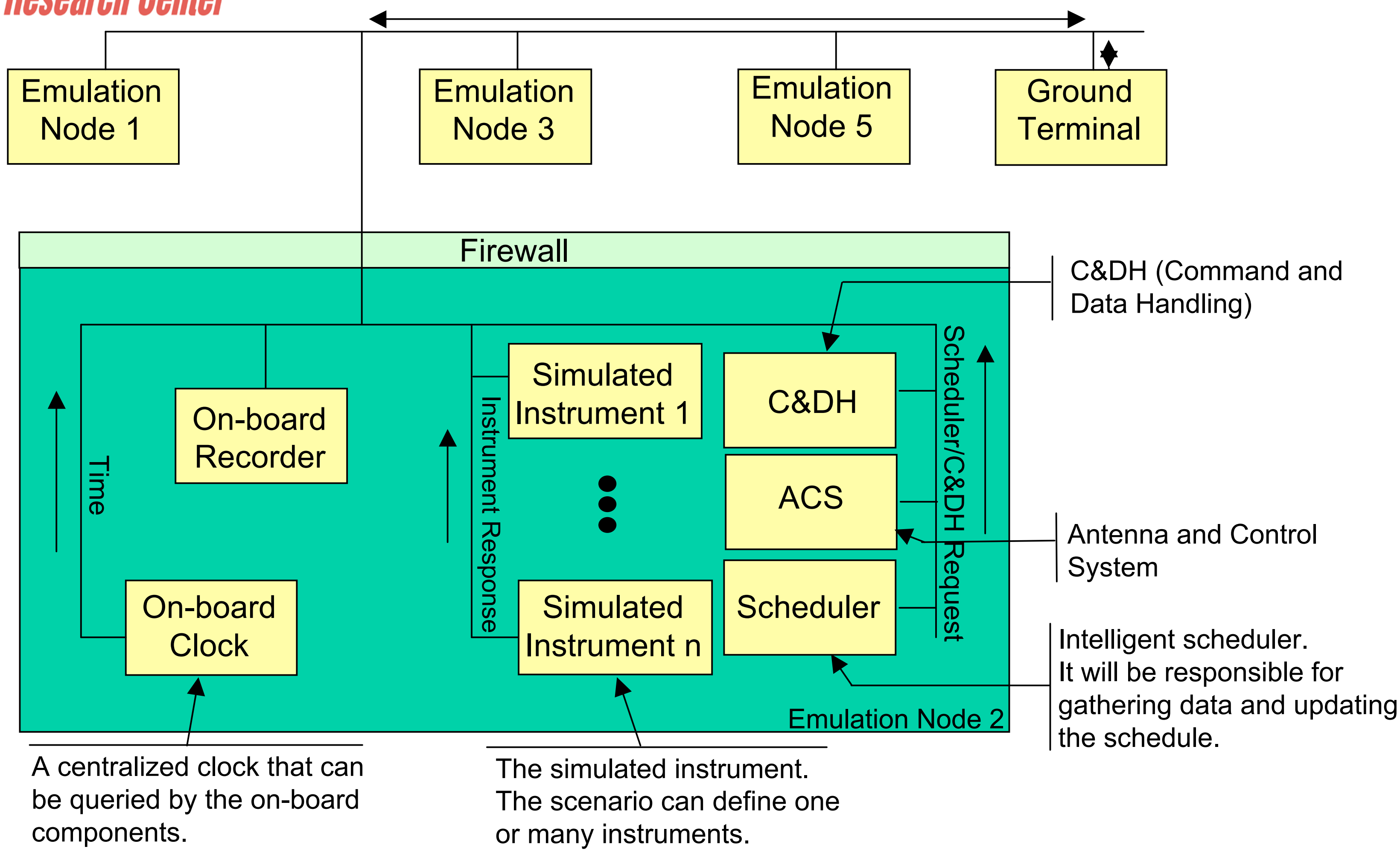
Simplified Integration using Common Tools

- SCEF developed utilities for researchers home environment.

SCEF Architecture



SCEF Node Architecture



Example Scenarios for SCEF

Mission Types
- LEO-based, GEO-based
- Constellations, Single Satellite Missions

Research Algorithms
- Scheduling Algorithms
- Command and Data Handling (C&DH)
- Antenna Control

Security
- Modifications to Firewalls and Routers
- IP Sec, VPNs

Communications
- Modifications to the TCP/IP Stack
- Throughput

Networking Issues
- Modifications to Routing Algorithms

Automated On-Board Schedule Revision

SCEF is working in coordination with the Automated Reasoning Group from Ames.

- A custom algorithm will replace the default on-board scheduler.

Produces a dynamic schedule for the science instruments

- Set of requests, constraints and sensing instruments.

Observation conditions can change dramatically

- Unexpected cloud cover
- Serendipitous events
- Changes in resource capabilities.

Satellite have limited communication times with ground sites.

- Infeasible to make all decisions on the ground and upload new schedules.

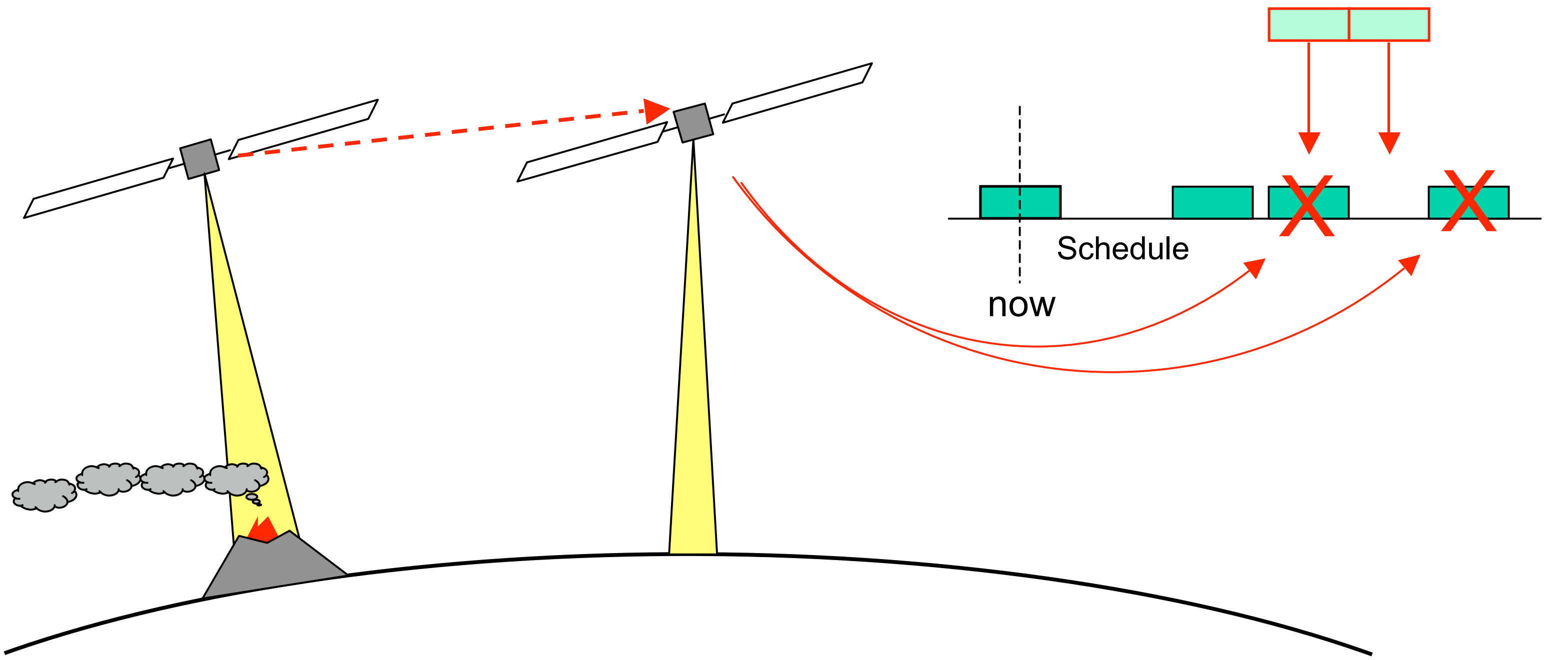
Perform some on-board decision making to modify schedule.

Automated On-Board Schedule Revision

Ames Research Center

Example: Using Intersatellite Communication to respond To Targets of Opportunity.

Replace future observations with ones of higher utility.



ESTC: June 23, 2004

Conclusions

Researchers from academia, government and industry will have access to a satellite emulation facility for modeling satellite missions.

SCEF serves two purposes:

- Models entire missions by defining scenarios that contain mission parameters.
- Integrates custom code into the environment to test algorithms for certain aspects of the mission.

Testbed uses open standards

- Linux
- TCP/IP

Serves the need for future missions.

- NASA is current designing complex future missions.
- Designed for both Earth-centric and Deep Space missions.

Interleaved Execution and Revision

For each time slot *t*
Consider the set *R* of requests that can be scheduled at time *t*.
Assign a heuristic value to each request in *R*.
While there are still requests to consider in *R*
choose *r* in *R* that has highest heuristic value
If SSR has sufficient capacity for *r*
acquire and assess the actual utility of *r*
Else if SSR has insufficient capacity for *r*
Let *W* be a set of past observations with combined utility less than that of *r* and whose combined SSR requirements + available space in SSR is sufficient for *r*
If W is not empty
Let *w* be a minimum utility in *W*
Discard *w* for SSR release
Acquire and assess actual utility of *r*
Else remove *r* from *R*

Hardware Architecture

SCEF contains 32 nodes and 2 controllers.

- Controller is responsible for starting the emulation and controlling the nodes.

Controllers

- Pentium III Class Machines (900 MHz)
- 4 GB Memory
- 234 GB On-line Storage
- Gigabit Interfaces

Nodes

- Pentium IV Class Machines (3.06 GHz)
- 1 GB Memory
- 80 GB On-line Storage
- Gigabit Inerfaces

SCEF Node Architecture

Ames Research Center

Each node represents a satellite or ground station.

Each component is modeled as a UNIX process.

- Components are customizable.
- Examples include Command and Data Handling (C&DH), Recorder, Simulated Instruments, On-Board Scheduler, Antenna and Control Systems

Open standards

- TCP/IP and Ethernet
- UNIX Operating Systems
    - Standardized on RedHat Fedora Core I

Third-party Software

- Satellite Toolkit (STK) for orbit generation
- NetSpec (U. of Kansas) for data throughput.

Assumptions and Implications

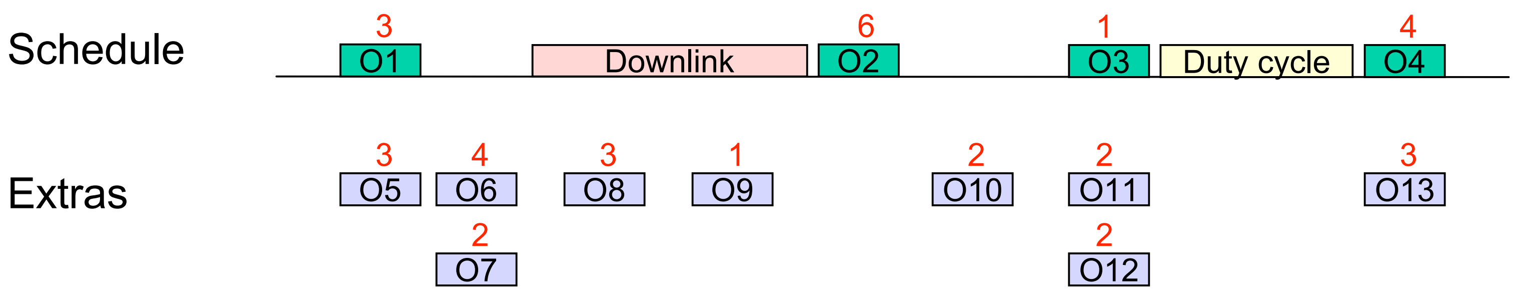
Ames Research Center

- Limited time for making on-board decisions
- Limited processing power
- Limited inputs
  - Existing schedule
  - Set of additional (desirable) observations
  - No knowledge of other satellites
- Updates on observation priority/utility

→ Schedule *Revision,* Not rescheduling

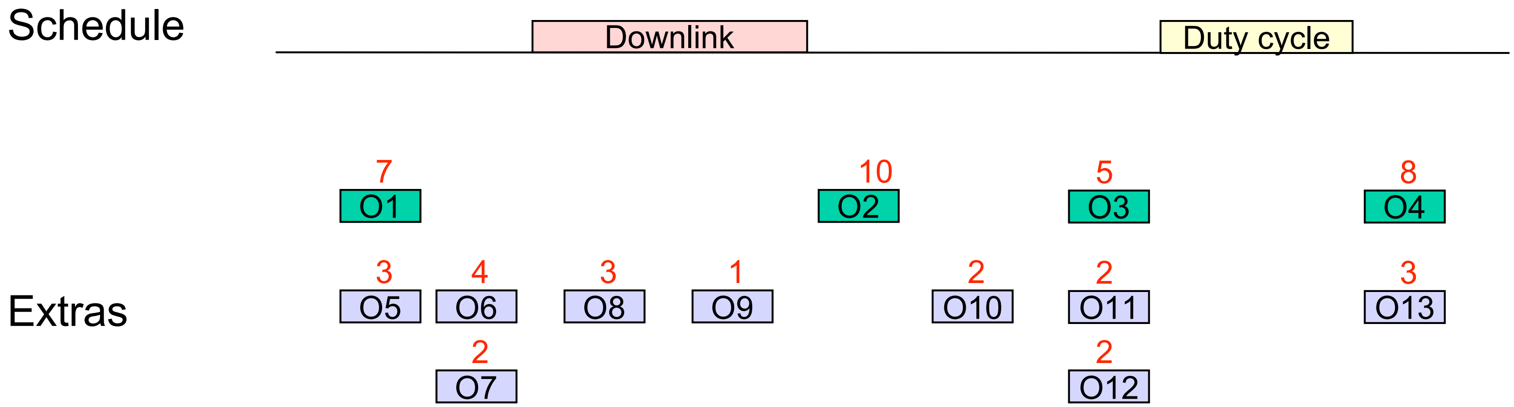
→ Existence of separate scheduling system

Schedule Revision System: Inputs



Set-up

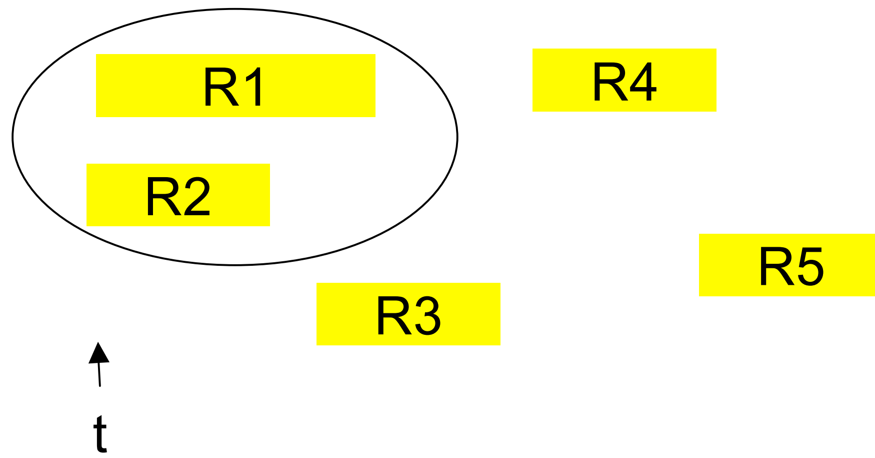
1. Elevate utility of scheduled observations.
2. Remove scheduled observations.



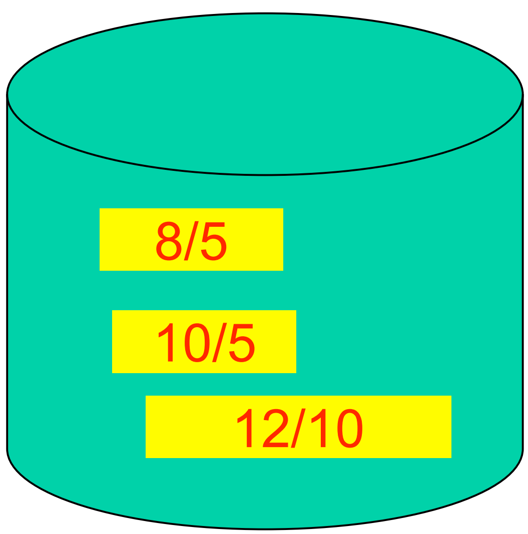
Allows system to revert to original schedule unless there is a change in utility.

Interleaved Execution and Revision

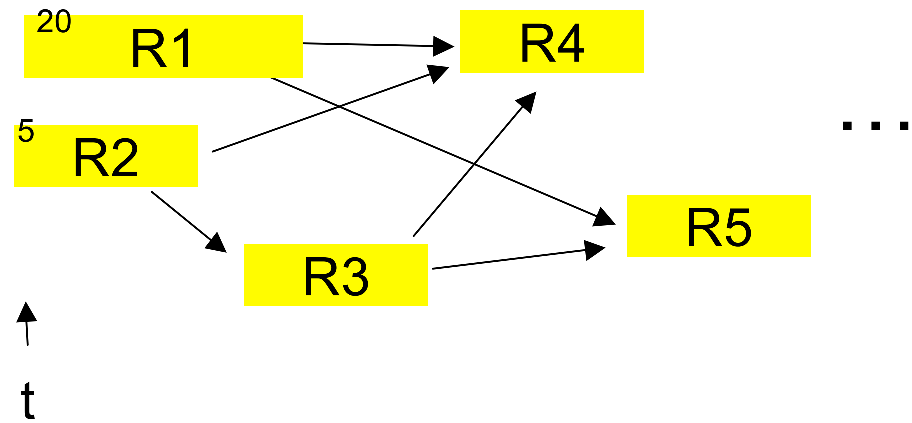
Ames Research Center



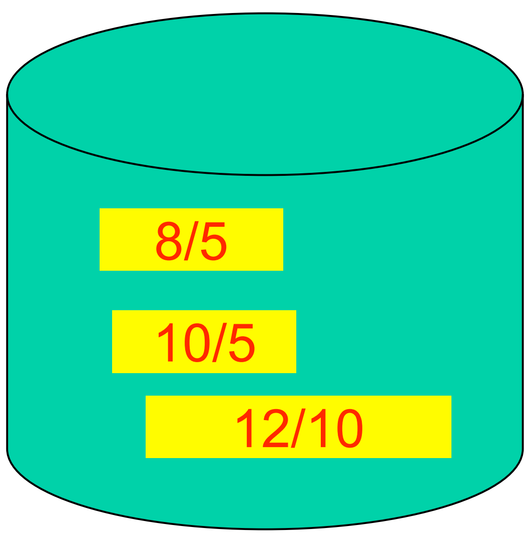
1. Identify candidate requests that can be scheduled at t.



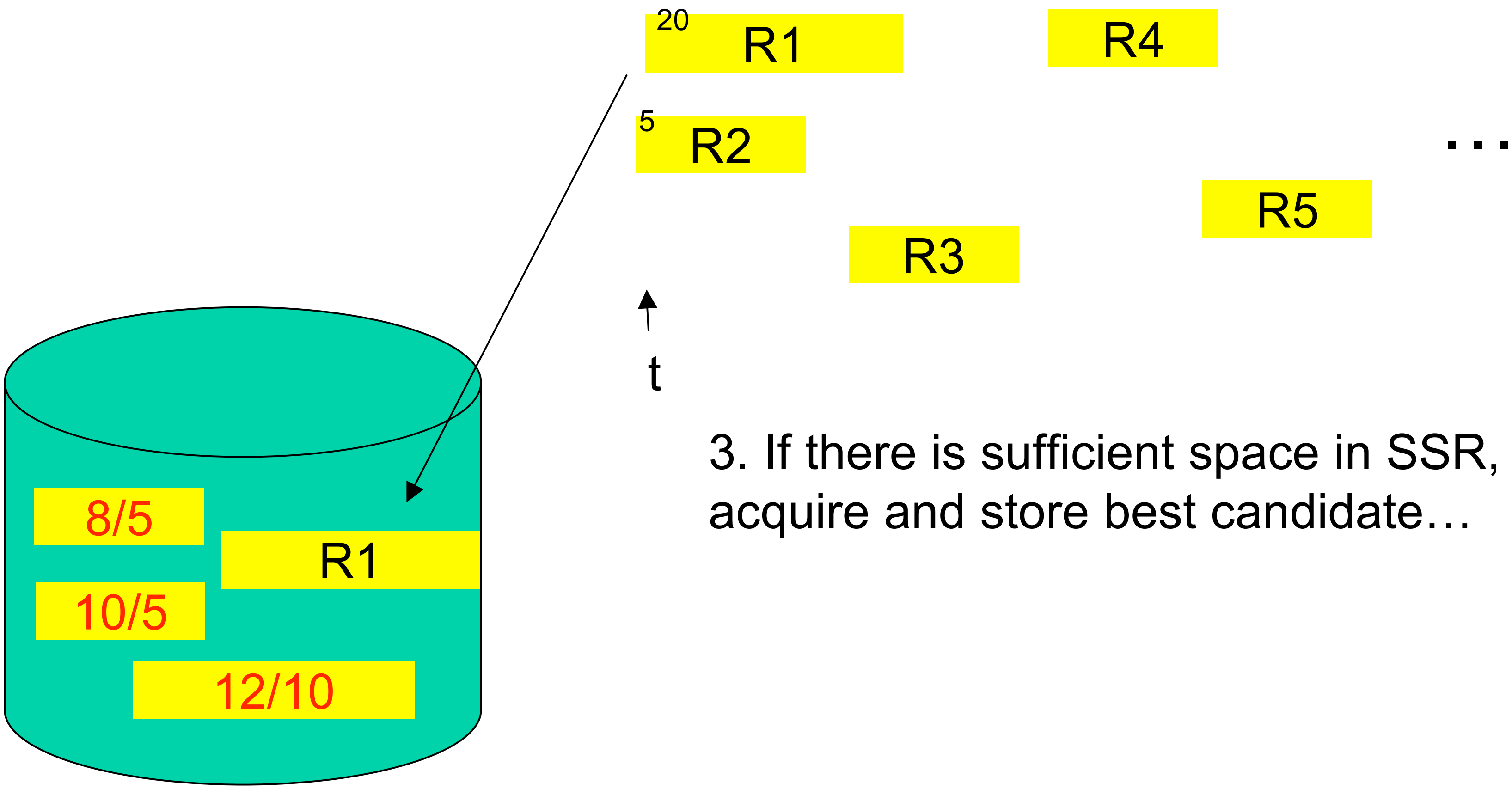
Interleaved Execution and Revision



2. Compute heuristic value of each candidate request

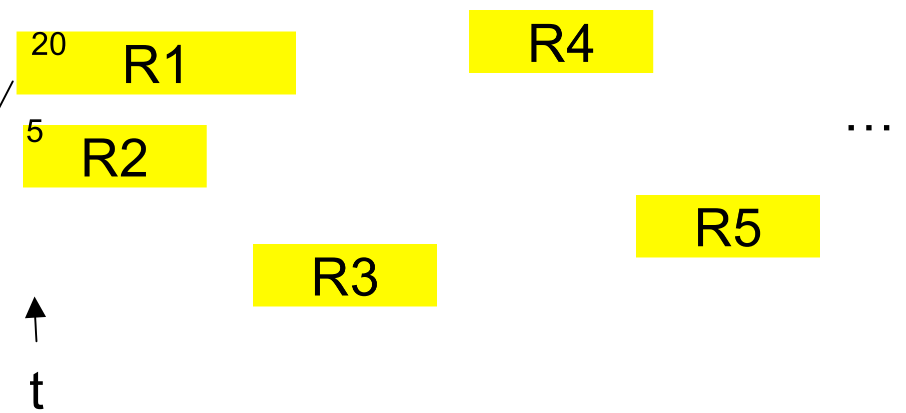


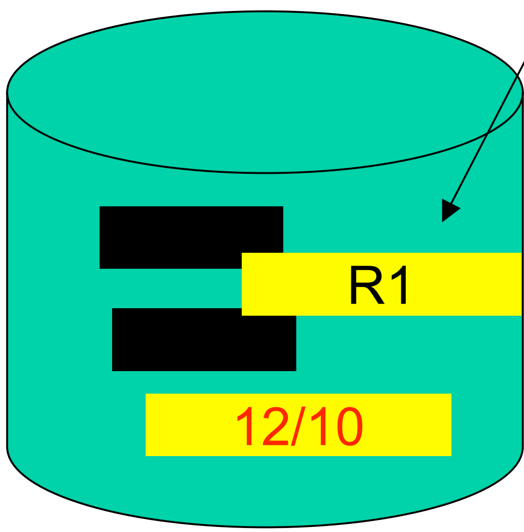
Interleaved Execution and Revision



3. If there is sufficient space in SSR, acquire and store best candidate…

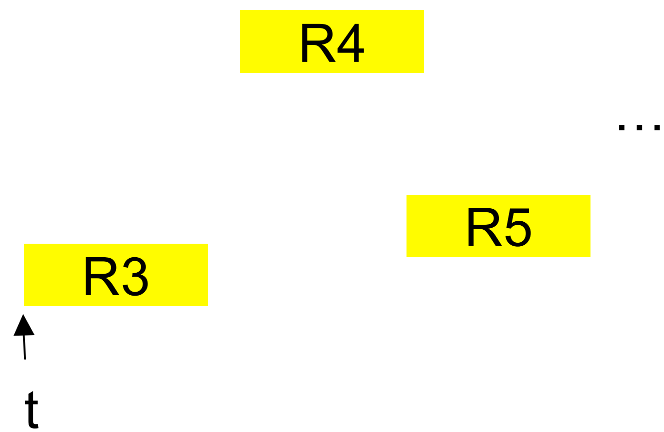
Interleaved Execution and Revision



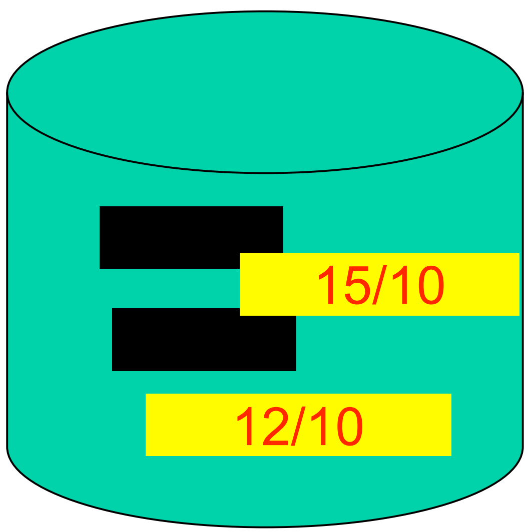


4. … else if there is a set of acquired images of lesser overall actual utility that can be removed, replace them with higher utility image

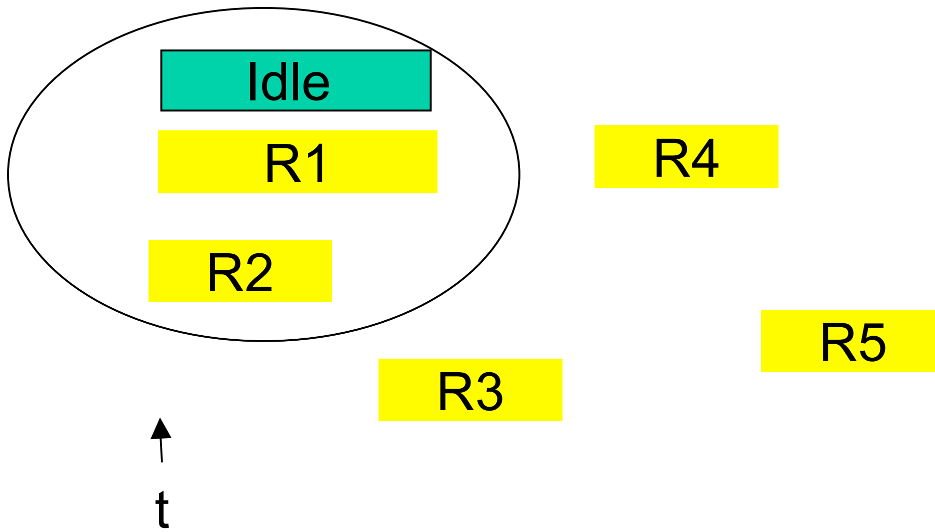
Interleaved Execution and Revision

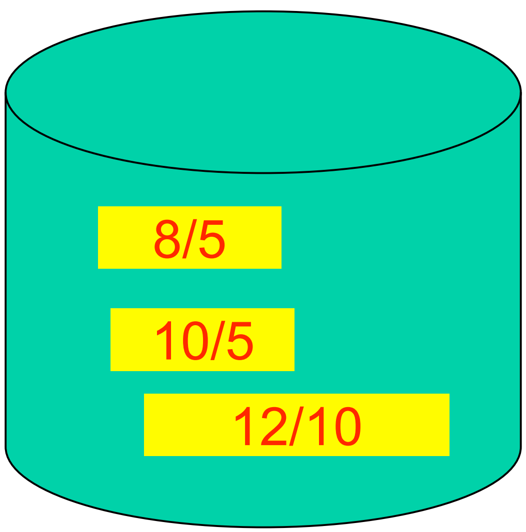


5. Assess actual utility of acquired image and repeat process.



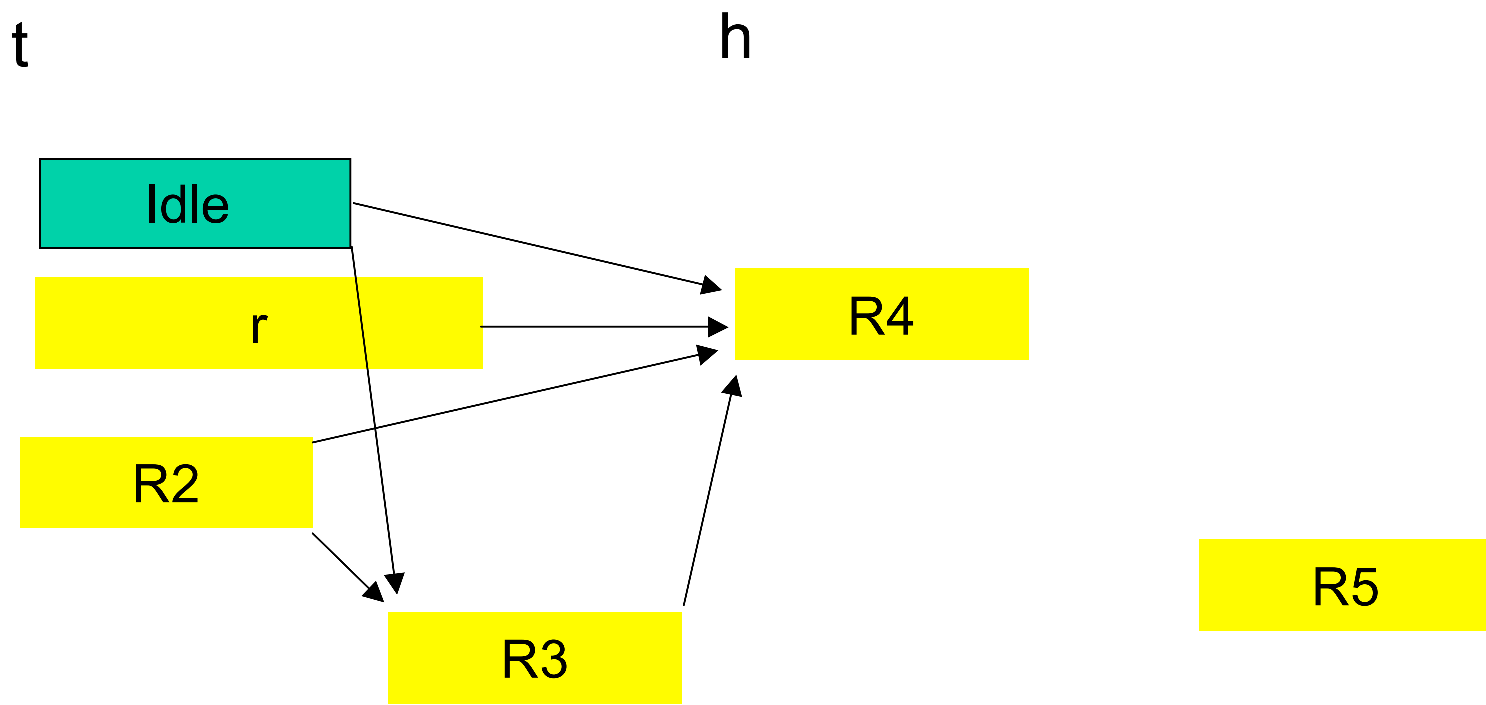
The “do nothing” option





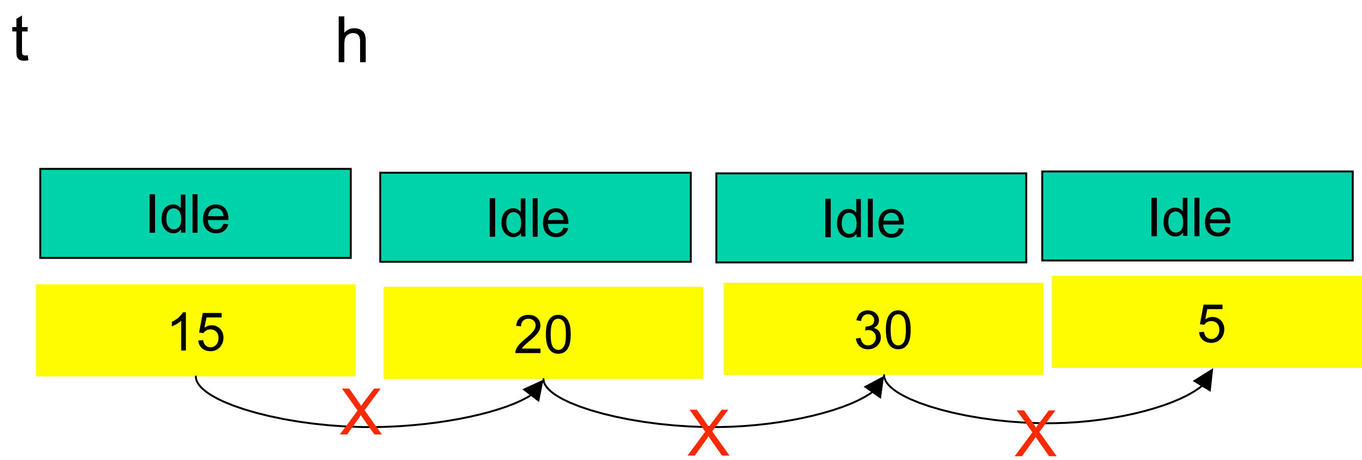
The best option at t might be slew instrument to take R3. Assuming no observing while slewing, best not to schedule anything at t.

Fixed Lookahead Strategy



For each request r, its heuristic value is expected overall utility of the best schedule starting with r and ending at a fixed lookahead horizon h.

Problem: Horizon Effect



With an insufficient lookahead horizon, heuristic will indefinitely prefer to do nothing if there are constraints prohibiting the taking of consecutive images. This may result in inferior schedules.

Variable Lookahead Strategy

Ames Research Center

- Idea: Expand lookahead horizon until there is an agreement between the last *k* horizons on the best candidate for the current level (or some maximum depth is reached).
- Example: k = 2 (works well in practice)
- When best candidate for H3 = best candidate for H2 = 15, algorithm chooses this outcome.

Idle	Idle	Idle	Idle
15	20	30	5

H0 = 15 H1 = Idle H2 = 15 H3 = 15